Securing SSH with public keys

Securing keys with YubiKey

Make sure you use on both your client and server at least version 8.2 of OpenSSH, because previous versions do not support 2FA with hardware security keys:

`ssh -V`

Check if OpenSSH can generate U2F keys:

`ssh-keygen --help`

and you should see something like that:

`[-t dsa | ecdsa | ecdsa-sk | ed25519 | ed25519-sk | rsa]`

The ecdsa-sk and ed25519-sk types are the only valid ones for 2FA enabled SSH key pairs. The ‘-sk’ part stands for ‘security key’.

To generate SSH key-pair:

`ssh-keygen -t ed25519-sk -f ~/.ssh/securitykey`

> If you get this or something similar error while trying to generate a key: "error while loading shared libraries: libfido2.so.1", you have to install the "libfido2" dependency on your system.

It’ll ask you to touch your YubiKey:

> Generating public/private ed25519-sk key pair.
> You may need to touch your authenticator to authorize key generation.

Just touch the metal circle and it’ll bind the SSH key pair to your YubiKey.

When it says “Enter passphrase (empty for no passphrase)”, you can just press enter to leave it empty.

> Beware "ed25519-sk" is only supported from version 5.2.3, so if you have a YubiKey with an earlier firmware, use ecdsa-sk

>

> To see which firmware version is on your YubiKey:
>
> https://www.yubico.com/support/download/yubikey-manager

Two keys are generated under ~/.ssh

- securitykey (private key, make sure to never share it)
- securitykey.pub (public key that you have to copy to your server)

To copy the public security key to the server you can either use:

`ssh-copy-id -i ~/.ssh/securitykey.pub user@server`

or

login to the server and create the `~/.ssh/authorized_keys` file if it doesn't exist, then copy and paste the content from your `securitykey.pub` into the `authorized_keys` file.

Finally to connect to the server:

`ssh -i securitykey user@server`

If it works, then you can make the **server more secure by disabling password authentication** .

Open `/etc/ssh/sshd_config` and find this section, comment it back and set it to no:

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
```

Then run:

`service ssh restart`

Securing keys with Trezor Model T

The steps are basically the same just like with YubiKey, except in this case `ed25519-sk` can be used as well.

Source:

- https://rameerez.com/how-to-use-yubikey-to-log-in-via-ssh-to-server/
- https://cryptsus.com/blog/how-to-configure-openssh-with-yubikey-security-keys-u2f-otp-authentication-ed25519-sk-ecdsa-sk-on-ubuntu-18.04.html
- https://stackoverflow.com/questions/20898384/disable-password-authentication-for-ssh

Revision #3
Created 2023-01-01 19:39:33 UTC by Tozo
Updated 2023-01-01 20:36:11 UTC by Tozo